

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-219774

(43)Date of publication of application : 18.08.1995

(51)Int.Cl.

G06F 9/38

G06F 9/38

G06F 11/00

G06F 15/16

(21)Application number : 06-013421

(71)Applicant : FUJITSU LTD

(22)Date of filing : 07.02.1994

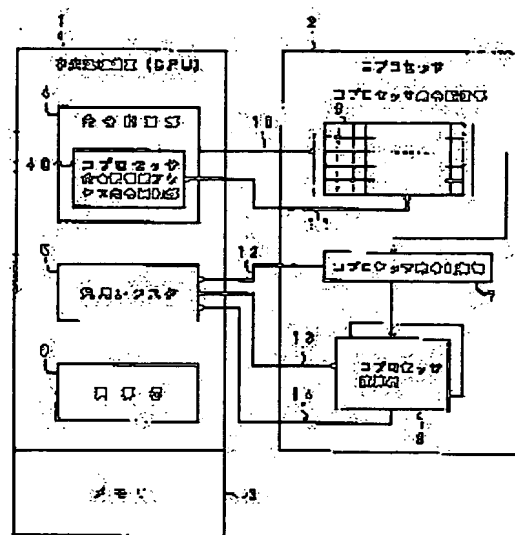
(72)Inventor : NISHIGAMI TOMOSHI

## (54) DATA PROCESSOR AND EXCEPTION PROCESSING METHOD

## (57)Abstract:

**PURPOSE:** To allow CPU to flexibly correspond to the generation of an exception and to verify the normal operation of a coprocessor at the time of the generation of the exception by providing a means for accessing a coprocessor instruction storage part in the coprocessor from CPU.

**CONSTITUTION:** This processor is provided with the means for accessing the processor instruction storage part 6 in the coprocessor 2 from CPU 1 and a group of instructions for executing the access. Then, the processor issues the instruction of reading out the contents of the processor instruction storage part 6 in the intended coprocessor 2 from the program of CPU 1 so as to inform the program of the contents of the specified coprocessor instruction storage part 6 through a general register 5, a memory 3, etc., so that the program can judge the contents and properly process it. The processor verifies the contents of the coprocessor instruction storage part 6 at the time of the generation of an exception like this and clears it or edits it to rewind if necessary, so as to retry, etc., a processor instruction.



## LEGAL STATUS

[Date of request for examination]

21.09.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision]

BEST AVAILABLE COPY

of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-219774

(43) 公開日 平成7年(1995)8月18日

(51) Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/38	3 8 0 B			
	3 7 0 C			
11/00	3 1 0 A			
15/16	3 7 0 Z			

審査請求 未請求 請求項の数 5 O L (全 8 頁)

(21) 出願番号 特願平6-13421  
 (22) 出願日 平成6年(1994)2月7日

(71) 出願人 000005223  
 富士通株式会社  
 神奈川県川崎市中原区上小田中1015番地  
 (72) 発明者 西上 知史  
 神奈川県川崎市中原区上小田中1015番地  
 富士通株式会社内  
 (74) 代理人 弁理士 小笠原 吉義 (外2名)

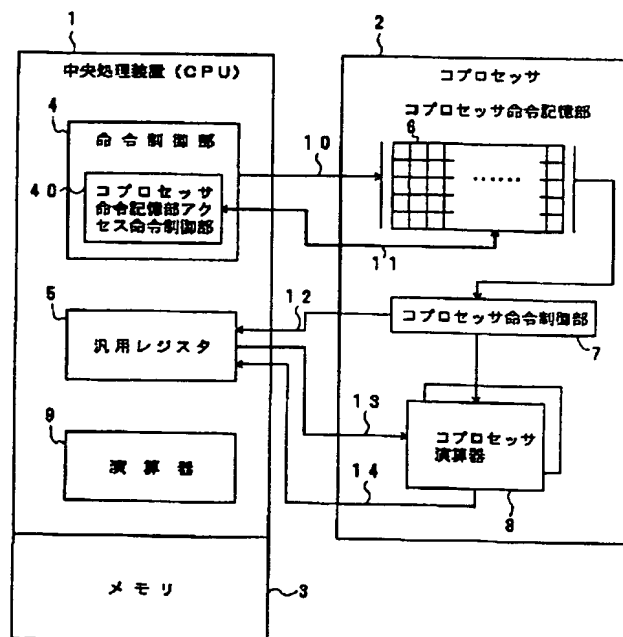
(54) 【発明の名称】 データ処理装置および例外処理方法

(57) 【要約】

【目的】 CPU 1 と、CPU 1 からのコプロセッサ命令を非同期的に実行するコプロセッサ 2 を備えたデータ処理装置に関し、特に例外発生時に例外処理を適確に行うための手段を提供することを目的とする。また、コプロセッサ 2 の正常動作を容易に検証できるようにすることを目的とする。

【構成】 CPU 1 のプログラムから、コプロセッサ命令記憶部 6 に対するアクセス命令を用いることにより、例外発生時にコプロセッサ命令記憶部 6 の内容を調べ、必要であれば命令をクリアしたり編集して書き戻すことにより、コプロセッサ命令の再試行等を行う。

本発明の原理説明図



## 【特許請求の範囲】

【請求項 1】 命令を実行する中央処理装置(1)と、その中央処理装置(1)に接続され、中央処理装置(1)から指示されたコプロセッサ命令を非同期的に実行する 1 個以上のコプロセッサ(2)とを有するデータ処理装置であって、前記コプロセッサ(2)内に中央処理装置(1)から指示されたコプロセッサ命令を記憶するコプロセッサ命令記憶部(6)と、コプロセッサ命令記憶部(6)に記憶されている命令の実行を制御するコプロセッサ命令制御部(7)と、コプロセッサ命令の演算を実行するコプロセッサ演算器(8)とを備えたデータ処理装置において、前記中央処理装置(1)は、前記コプロセッサ(2)におけるコプロセッサ命令記憶部(6)の内容を参照または更新する命令の命令実行制御手段(40)を持ち、コプロセッサ命令記憶部(6)の内容を中央処理装置(1)で動作するプログラムから前記命令によりアクセス可能に構成されたことを特徴とするデータ処理装置。

【請求項 2】 請求項 1 記載のデータ処理装置において、前記コプロセッサ命令記憶部(6)は 1 つの命令キューまたはコプロセッサ命令の種類に応じた複数の命令キューからなり、前記中央処理装置(1)がコプロセッサ命令記憶部(6)にアクセスする命令は、コプロセッサ命令記憶部(6)内の位置を指定する手段を持つことを特徴とするデータ処理装置。

【請求項 3】 請求項 1 または請求項 2 記載のデータ処理装置において、前記中央処理装置(1)がコプロセッサ命令記憶部(6)にアクセスする命令として、コプロセッサ命令記憶部(6)の一部または全部をクリアする命令を有することを特徴とするデータ処理装置。

【請求項 4】 請求項 1、請求項 2 または請求項 3 記載のデータ処理装置による例外処理方法において、前記中央処理装置(1)は、前記コプロセッサ(2)内でコプロセッサ命令の実行に関する例外が発生した場合に、例外発生時点で実行中であったもの以降の後続のコプロセッサ命令をキャンセルして例外処理を行うことを特徴とする例外処理方法。

【請求項 5】 請求項 1、請求項 2 または請求項 3 記載のデータ処理装置による例外処理方法において、前記中央処理装置(1)は、前記コプロセッサ命令記憶部(6)にアクセスする命令を用いたプログラムにより、自分で発行したコプロセッサ命令の内容を確認し、さらにその内容に対して行った変更を確認することによって、前記コプロセッサ(2)における各部の正常な動作を検証することを特徴とする例外処理方法。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、命令を高速で実行するために、命令実行を非同期的に処理するコプロセッサを持つデータ処理装置および例外処理方法に関する。

【0002】 例えば、浮動小数点演算装置などのコプロ

セッサを持つデータ処理装置において、例外処理を適確に行うためには、コプロセッサ内の演算器で発生した例外の情報を必要に応じてプログラムに対し詳細に報告しなければならない。また、例外発生時において予測される動作が正しく行われるかどうかを検証するためには、異常状態をプログラムによって設定し、その結果を確認する手段が必要になる。

## 【0003】

【従来の技術】 従来の非同期型コプロセッサの例外処理システムとして、例えば特開平 4-106652 号公報に示されるようなものがある。このような従来の例外処理システムにおいては、コプロセッサから中央処理装置(以下、CPU という)に対して、例外が発生したコプロセッサ命令に関する直接の情報だけが通知されていた。なお、ここで例外の発生とは、例えばある値を 0 で除算するとか、命令中のオペランドでアクセスできない領域を指定するというような命令を正常に実行できない事象が発生することをいう。

【0004】 図 6 は、その従来技術の構成を示すブロック図である。図 6 中、60 は CPU、61 は命令制御部、62 は汎用レジスタ、63 は演算器、64 はメモリ、70 はコプロセッサ、71 は制御部、72 は単数または複数のコプロセッサ演算器、73 はコプロセッサ命令キュー、74 は例外フラグ用レジスタである。81 は命令制御部 61 からコプロセッサ命令キュー 73 にコプロセッサ命令を通知するコプロセッサ命令バス、82 は制御部 71 から汎用レジスタ 62 へ書き込みレジスタ番号および読み出しレジスタ番号等を通知するレジスタ番号バス、83 は汎用レジスタ 62 からコプロセッサ演算器 72 にソースデータを通知するソースデータバス、84 はコプロセッサ演算器 72 の出力を汎用レジスタ 62 に通知する出力バスを表す。

【0005】 コプロセッサ演算器 72 で演算処理中に例外が発生した場合、コプロセッサ命令キュー 73 内に設けた例外フラグ用レジスタ 74 に例外の種類を書き込み、コプロセッサ 70 内の制御部 71 は、コプロセッサ命令キュー 73 での例外発生位置を認識し、例外フラグ用レジスタ 74 から例外の種類を判別して例外の種類に応じた例外処理を行う。そして、制御部 71 は、CPU 60 に対して例外が発生したコプロセッサ命令に関する情報を通知する。

## 【0006】

【発明が解決しようとする課題】 以上のような従来の技術では、例外が起こったコプロセッサ命令の情報だけしか CPU には報告できない。したがって、例外発生時点における一連の命令によるコプロセッサ内の処理の流れが容易には把握できないという問題があった。また、CPU からコプロセッサに対して指示されたコプロセッサ命令が、コプロセッサ命令を記憶するキュー内に正常に保持されているかどうかの検証も同様に困難であった。

さらに、例外処理から復帰する場合に、CPUがコプロセッサ命令を記憶するキュー内の情報を任意に変更することができず、CPUにおける柔軟な対応が望めなかった。

【0007】本発明は上記問題点の解決を図り、コプロセッサ内のコプロセッサ命令を記憶する部分をCPUからアクセスするための手段となるハードウェアおよびそれを実行するための命令群を提供することにより、例外発生時におけるCPUの柔軟な対応およびコプロセッサの正常動作の検証を可能にすることを目的とする。

【0008】

【課題を解決するための手段】図1は、本発明の原理説明図である。図1において、1は中央処理装置(CPU)、2はコプロセッサ、3はメモリ、4は命令制御部、40はコプロセッサ命令記憶部アクセス命令制御部、5は演算対象となるデータを保持する汎用レジスタ、6はコプロセッサ命令記憶部、7はコプロセッサ命令制御部、8はコプロセッサ演算器、9は命令で指定された加減算等の演算を行う演算器、10はコプロセッサ命令バス(以下、命令バスという)、11はコプロセッサ命令記憶部アクセスバス(以下、アクセスバスという)、12は汎用レジスタ番号/割込み情報バス(以下、レジスタ番号バスという)、13は汎用レジスタ読み出しデータバス(以下、読み出しデータバスという)、14は汎用レジスタ書き込みデータバス(以下、書き込みデータバスという)を表す。

【0009】命令制御部4は、メモリ3に格納されているプログラム命令を同期的に順次読み出してシーケンシャルに実行し、これら一連の命令の実行中に、コプロセッサ2に演算させるべき命令があれば、命令バス10を介してコプロセッサ2へ通知する制御手段である。コプロセッサ命令記憶部アクセス命令制御部40は、アクセスバス11を介してコプロセッサ2におけるコプロセッサ命令記憶部6の内容を参照または更新する命令の命令実行制御手段である。

【0010】コプロセッサ命令記憶部6は、CPU1から指示されたコプロセッサ命令を記憶する手段である。コプロセッサ命令制御部7は、コプロセッサ命令記憶部6に記憶されている命令の実行を制御し、レジスタ番号バス12を介して汎用レジスタ番号、割り込み情報等をCPU1へ通知する手段である。

【0011】コプロセッサ演算器8は、汎用レジスタ5から読み出しデータバス13を介して通知されたデータについてコプロセッサ命令で指定された演算を行い、その結果を書き込みデータバス14を介して汎用レジスタ5へ出力する手段である。

【0012】命令バス10は、命令制御部4からコプロセッサ命令記憶部6へコプロセッサ命令を通知するバスである。アクセスバス11は、コプロセッサ命令記憶部アクセス命令制御部40からコプロセッサ命令記憶部6

へのアクセスに使用されるバスである。

【0013】レジスタ番号バス12は、コプロセッサ命令制御部7から汎用レジスタ5へ汎用レジスタ番号または割込み情報を通知するバスである。読み出しデータバス13は、汎用レジスタ5から読み出したデータをコプロセッサ演算器8へ通知するバスである。書き込みデータバス14は、コプロセッサ演算器8の演算結果を汎用レジスタ5へ通知するバスである。

【0014】本発明では、コプロセッサ命令記憶部6の内容を参照または更新するコプロセッサ命令記憶部アクセス命令が用意され、この命令の実行によって、コプロセッサ命令記憶部6の内容を、CPU1で動作するプログラムからアクセス可能になっている。

【0015】コプロセッサ命令記憶部6は、1つの命令キューまたはコプロセッサ命令の種類に応じた複数の命令キューからなり、コプロセッサ命令記憶部アクセス命令は、オペランドによってその命令キューにおけるアクセス対象の位置を指定することができる。このコプロセッサ命令記憶部アクセス命令として、コプロセッサ命令記憶部6の内容をCPU1内のプログラムがアクセス可能な汎用レジスタ5(またはメモリ3)にロード/ストアする命令がある。また、このコプロセッサ命令記憶部アクセス命令の一つとして、コプロセッサ命令記憶部6の一部または全部をクリアする命令がある。

【0016】CPU1は、コプロセッサ2内でコプロセッサ命令の実行に関する例外が発生した場合に、例外発生時点で実行中であったもの以降の後続のコプロセッサ命令を、例えばコプロセッサ命令記憶部6の内容を編集することによりキャンセルして例外処理を行う。

【0017】また、CPU1は、コプロセッサ命令記憶部アクセス命令を用いることにより、自分で発行したコプロセッサ命令の内容を確認し、さらにその内容に対して行った変更を確認することによって、コプロセッサ2における各部の正常な動作を検証するプログラムを動作させる。

【0018】以上のように、本発明は、コプロセッサ2内のコプロセッサ命令記憶部6をCPU1からアクセスするための手段と、コプロセッサ命令記憶部6へのアクセスを実行するための新たな命令群とを提供する。

【0019】

【作用】CPU1のプログラムから目的とするコプロセッサ2内にあるコプロセッサ命令記憶部6の内容を読み出す命令を発行することにより、指定されたコプロセッサ命令記憶部6の内容が汎用レジスタ5またはメモリ3等を通じてプログラムに通知される。プログラムはその内容を判断して、適切な処理を行うことができる。さらに、コプロセッサ命令記憶部6の内容を直接、または内容を読み出してからそれを編集して書き戻すことができるので、コプロセッサ2の動作、特にコプロセッサ命令記憶部6の動作の正常性を、診断プログラム等によって

容易に検証することができる。

#### 【0020】

【実施例】以下、本発明の実施例を図面を用いて詳細に説明する。図2は本発明の一実施例を示す構成図である。実施例としてCPU内部に同期的に整数演算命令等の処理を行う演算器6があり、非同期的に動作するコプロセッサとして浮動小数点演算器33を持つ浮動小数点演算装置30が備えられている構成を用いて説明するが、非同期的に動作するコプロセッサであれば同様に動作可能である。例えば、コプロセッサは、ベクトル演算装置や特殊な整数演算器を有するものなどであってもよい。

【0021】図2において、1は中央処理装置(CPU)、3はメモリ、30は浮動小数点演算装置、5は汎用レジスタ、6は演算器、10は命令バス、12はレジスタ番号バス、13は読み出しデータバス、14は書き込みデータバス、21は命令アドレス発生装置、22は命令制御装置、23は一般命令制御部、24はコプロセッサ命令制御部、25は命令キューアクセス命令制御部、31は浮動小数点命令キュー、32は浮動小数点命令制御部、33は浮動小数点演算器、34は命令コードレジスタ、35は第1オペランド・レジスタ、36は第2オペランド・レジスタ、37は第3オペランド・レジスタ、38は例外フラグ用レジスタ、41は浮動小数点命令キュー番号バス(以下、命令キュー番号バスという)、42は浮動小数点命令キュー読み出しデータバス(以下、命令キュー読み出しデータバスという)、43は浮動小数点命令キュー書き込みデータバス(以下、命令キュー書き込みデータバスという)を表す。

【0022】命令制御装置22は、命令アドレス発生装置21から命令のアドレスを得て、メモリ3に格納されたプログラム命令を順次読み出し、その命令実行の制御を行う装置である。読み出された命令は、まず一般命令制御部23においてデコードされ、それが通常の整数演算命令等であれば、CPU1の演算器6により実行され、結果が汎用レジスタ5またはメモリ3等へ書き込まれる。

【0023】命令が浮動小数点演算装置30に実行されるべき浮動小数点演算命令である場合には、コプロセッサ命令制御部24は、命令バス10を介してその浮動小数点演算命令を浮動小数点命令キュー31へ送る。

【0024】浮動小数点命令キュー31の各エントリは、浮動小数点演算命令の命令コードを格納する命令コードレジスタ34、命令の第1オペランドを指定する第1オペランド・レジスタ35、命令の第2オペランドを指定する第2オペランド・レジスタ36、演算結果を格納する第3オペランドを指定する第3オペランド・レジスタ37、例外フラグを格納する例外フラグ用レジスタ38から構成される。これらの使い方は、特開平4-106652号公報等々に示されている従来技術と同様であ

る。

【0025】浮動小数点命令制御部32は、浮動小数点命令キュー31から順次命令を読み出し、命令コードに応じて浮動小数点演算器33を制御する。浮動小数点演算器33は、読み出しデータバス13を介して第1オペランドおよび第2オペランドで指定された汎用レジスタ5から得たデータについて演算を実行し、その演算結果を書き込みデータバス14を介して第3オペランドで指定された汎用レジスタ5へ書き込む。

【0026】このような浮動小数点演算装置30の動作中に例外が発生すると次のように動作する。浮動小数点演算器33は、例外がどのような種類の例外であったかを例外フラグ用レジスタ38へ書き込む。浮動小数点命令制御部32は、浮動小数点演算器33での例外を検出するとCPU1に例外発生を通知する。これにより、CPU1の命令制御装置22は特定の例外割込みを発生させて例外処理ルーチンを起動する。

【0027】命令キューアクセス命令制御部25は、一般命令制御部23により判読した命令が浮動小数点命令キュー31にアクセスする命令であるときに起動される。命令キューアクセス命令制御部25は、命令キュー番号バス41、命令キュー読み出しデータバス42、命令キュー書き込みデータバス43を用いて、浮動小数点命令キュー31の内容の読み出し、書き出し、クリア等の処理を制御する。

【0028】本発明の実施例において、コプロセッサ命令キュー(浮動小数点命令キュー31等)にアクセスするために用意された命令の例を、図3に示す。図3において(A)、(B)、(C)はコプロセッサが浮動小数点演算装置の場合の例であるが、他の例としてコプロセッサが整数演算装置である場合の命令の例についても(D)、(E)に示す。

【0029】図3の(A)は命令キュークリア命令、(B)は命令キューへのムーブ命令、(C)は命令キューからのムーブ命令、(D)は命令キューへのロード命令、(E)は命令キューからのストア命令である。特に、ここでは(B)、(C)は汎用レジスタとのmove型命令、(D)、(E)はメモリとの間のload/store型命令の例を示しているが、もちろんコプロセッサの演算装置の種類およびアクセス命令の種類によって、別の命令セットを用いることも可能である。命令中、'GRi'は4バイト長のi番目の汎用レジスタ、'FQj'は32段・4バイト長または8バイト長で構成される浮動小数点命令キューのj番目のエントリを表す。また、'NQk'はコプロセッサが特殊な整数演算を実行する整数演算器を持つものである場合の4バイト長で構成される整数演算命令キューのk番目のエントリを表す。これらのバイト長は一例であるので、本発明はこれに限られるわけではない。

【0030】図3(A)の命令キュークリア命令“c

1.  $f q \text{ GR } i$  は、 $i$  番目の汎用レジスタ  $GR i$  の値中で '1' の立っているアドレスに対応する位置の 'FQj' をクリアする命令である。すなわち、 $GR i$  における第0ビット目が '1' であれば、命令キューの先頭エントリを、第1ビット目が '1' であれば、2番目のエントリを、...というように該当するエントリをクリアする。

【0031】図3 (B) の命令キューへのムーブ命令 " $mv. \text{ gr } f q \text{ GR } i, FQj$ " は、'GRi' と 'GRi+1' の内容を8バイト長の 'FQj' へ転送する命令である。

【0032】図3 (C) の命令キューからのムーブ命令 " $mv. f q \text{ gr } FQj, GR i$ " は、8バイト長の 'FQj' の内容を 'GRi' と 'GRi+1' へ転送する命令である。

【0033】図3 (D) の命令キューへのロード命令 " $ldnq \text{ GR } i, GR j, NQk$ " は、'GRi' と 'GRj' とで求められるアドレスの語データを 'NQk' にロードする命令である。

【0034】図3 (E) の命令キューからのストア命令 " $stnq \text{ NQk, GR } i, GR j$ " は、'NQk' の語データを 'GRi' と 'GRj' とで求められるアドレスにストアする命令である。

【0035】(D) および (E) の命令は、コプロセッサが整数演算を実行する整数演算器を持つものである場合などに用意される。図4は、実施例における例外処理のフローチャートを示す。図4の左側は中央処理装置

(CPU) による処理、右側はコプロセッサ (浮動小数点演算装置) による処理を示す。CPUとコプロセッサとは非同期で動作する。

【0036】(1) CPU1は読み出した命令中にコプロセッサ命令があると、そのコプロセッサ命令を発行してコプロセッサ2に通知し、コプロセッサ2を起動する。

(2) コプロセッサ2はコプロセッサ命令記憶部6 (キュー) に命令を入れ、命令を順次実行する。

【0037】(3) CPU1は、コプロセッサ命令をコプロセッサ2に引き渡した後、その完了を待たずに次の命令を実行する。

(4) コプロセッサ2において例外が発生すると、CPU1に対し例外発生を通知し、コプロセッサ命令記憶部6 (キュー) の対応する位置 (例外フラグ用レジスタ38) に例外情報を格納する。その後、コプロセッサ2は命令実行状態から保留状態に移す。

【0038】(5) CPU1は特定の例外割込みを発生させ、プログラムの処理を割込みハンドラに移す。例外処理ルーチンが以降の処理を行う。

(6) 例外が発生したコプロセッサ2内のコプロセッサ命令記憶部6 (キュー) の内容を読み出し、処理を継続できる例外か継続できない例外かを、例外の種類により判断する。

【0039】(7) ① 例えば、ある値を0で除算するというような不当な命令処理が発行されていた場合には、プロセスを異常終了させる。

(7) ② 命令自体に問題はない場合、コプロセッサ命令を再実行 (リトライ) させ、リトライ等でコプロセッサ2の固定障害と判断した場合には、システムをダウンさせる (この場合はここで処理は終了する)。

【0040】(8) コプロセッサ2内のコプロセッサ命令記憶部6 (キュー) の内容をクリアして、次の命令の実行に備える。

(9) コプロセッサ2は、コプロセッサ命令記憶部6 (キュー) の内容を指定に従ってクリアする (場合によっては一部を書き換える)。コプロセッサ2は待機状態または実行状態となる。

【0041】(10) CPU1は例外処理から復帰して、次の処理を行う。

次に、処理装置の保守点検等のために、動作確認用テストプログラムで命令キューアクセス命令を使用する場合の例を説明する。図5はコプロセッサ2の動作テスト処理方法のフローチャートを示す図である。ここでは、正しく書かれた通常のコプロセッサ命令では例外が発生しないので、まず正常系のテストを行った後に、プログラムで故意に例外を発生させ、例外発生に伴う動作の正当性を評価する。

【0042】まず、ステップS1において、例外が発生しない通常の処理で、正常系の動作を確認する。ステップS2において、異常が発生したかどうかを判断する。異常が発生した場合には、ステップS8で障害通知の処理を行い、異常が発生しない場合には、次のステップS3の処理へ進む。

【0043】ステップS3において、故意に不当な命令を発行することにより例外を起こし、期待した例外と割込みが発生することを確認する。ステップS4において、期待した例外と割込みが発生したかどうかを調べ、期待した例外と割込みが発生しなかった場合には異常であるので、ステップS8で障害通知の処理を行う。期待した例外と割込みが発生した場合には、次のステップS5の処理へ進む。

【0044】ステップS5において、例外割込みで停止しているコプロセッサ2のコプロセッサ命令記憶部6 (キュー) に対して命令キューアクセス命令を用いて読み書きを行い、内容を確認する。

【0045】ステップS6において、内容に異常があるかどうかを判断し、異常がある場合には、ステップS8で障害通知の処理を行う。異常がない場合には、ステップS7の処理へ進む。

【0046】ステップS7において、正常動作を確認した旨のメッセージを出力し、処理を終了する。

【0047】

【発明の効果】以上説明したように、本発明によれば、

例えばCPU上で動作するプログラム、特に割込みハンドラは、例外発生時に非同期型コプロセッサに対して、以下のような処理を行うことが可能となる。

【0048】① 割込みの種類を判断し、コプロセッサ命令記憶部に命令が残る可能性のない割込みの場合にはコプロセッサ命令記憶部に対する操作は行わない。

② 各コプロセッサ命令記憶部に命令が残っているかどうかを調べ、残っていなければコプロセッサ命令記憶部に対する操作は行わない。

【0049】③ コプロセッサ命令記憶部に命令の残余が存在した場合、割込みがそのコプロセッサからのものでなければ、単にコプロセッサ命令記憶部の内容を退避／復旧するだけで例外処理を済ませる。

【0050】④ コプロセッサ命令記憶部に命令が残っているコプロセッサからの割込みである場合、発生した例外の種類に応じて、退避したコプロセッサ命令記憶部の内容を編集し、割込みからの復旧時に復元する。

【0051】このような割込み処理を行う場合、コプロセッサ命令記憶部の内容全体を把握することにより、より適切な判断に基づいた処理を行うことが可能となる。また、任意の時点のコプロセッサ命令記憶部の内容を調査できるので、プログラムの開発における処理手順の最適化に有効な情報を得ることができる。

【0052】さらに、コプロセッサ命令記憶部の内容をプログラムから直接アクセスできるので、故障検出を自動的に行うテスト・プログラムを容易に作成できるようになり、それを用いることにより、障害発見を効率よく行うことが可能になる。

【図面の簡単な説明】

【図3】

アクセス命令の例

(A) 命令キュークリア命令

cl. fq GRI

(B) 命令キューへのムーブ命令

mv. grfq GRI, FQJ

(C) 命令キューからのムーブ命令

mv. fqgr FQJ, GRI

(D) 命令キューへのロード命令

ldnq GRI, GRJ, NQk

(E) 命令キューからのストア命令

stnq NQk, GRI, GrJ

【図1】本発明の原理説明図である。

【図2】本発明の実施例における構成を示す図である。

【図3】本発明の実施例によるアクセス命令の例を示す図である。

【図4】実施例における例外処理のフローチャートである。

【図5】コプロセッサの動作テスト処理方法のフローチャートである。

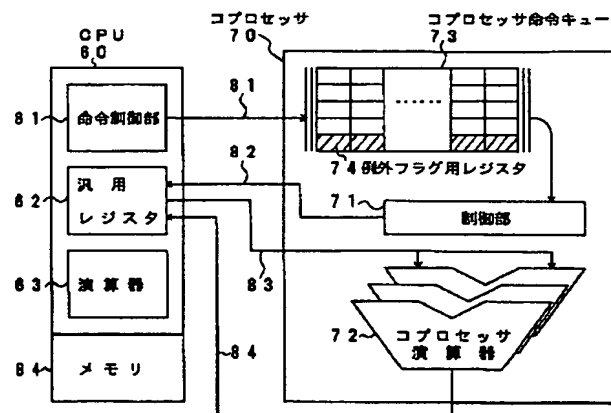
【図6】従来技術の例を示す図である。

【符号の説明】

- 1 中央処理装置 (CPU)
- 2 コプロセッサ
- 3 メモリ
- 4 命令制御部
- 5 汎用レジスタ
- 6 コプロセッサ命令記憶部
- 7 コプロセッサ命令制御部
- 8 コプロセッサ演算器
- 9 演算器
- 10 コプロセッサ命令バス (命令バス)
- 11 コプロセッサ命令記憶部アクセスバス (アクセスバス)
- 12 汎用レジスタ番号／割込み情報バス (レジスタ番号バス)
- 13 汎用レジスタ読み出しデータバス (読み出しデータバス)
- 14 汎用レジスタ書き込みデータバス (書き込みデータバス)
- 40 コプロセッサ命令記憶部アクセス命令制御部

【図6】

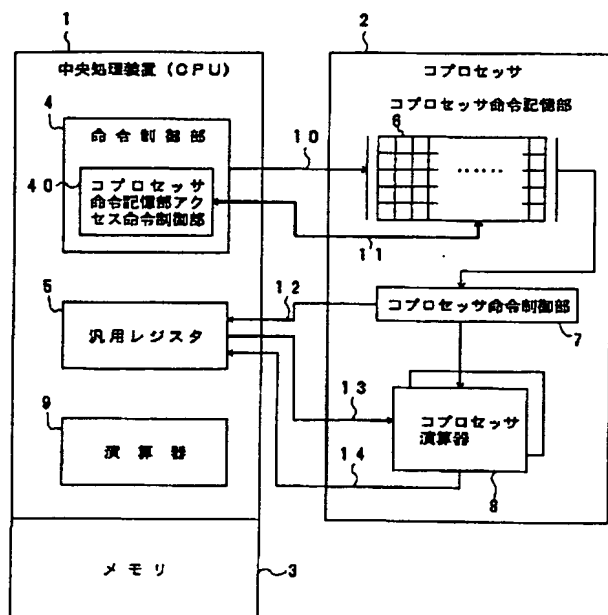
従来技術の例





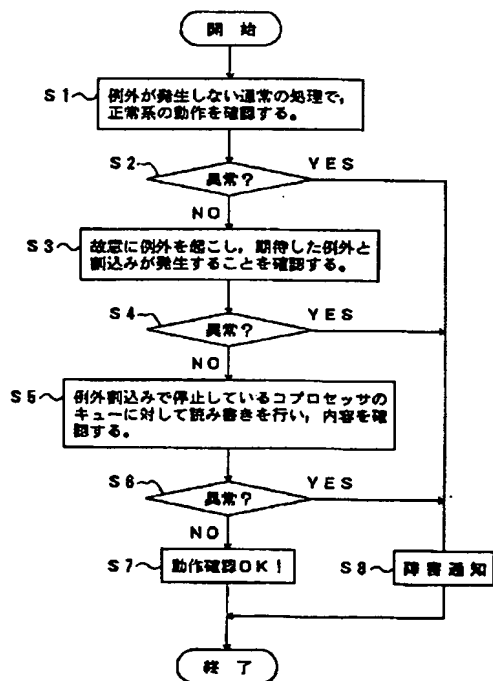
【図1】

本発明の原理説明図



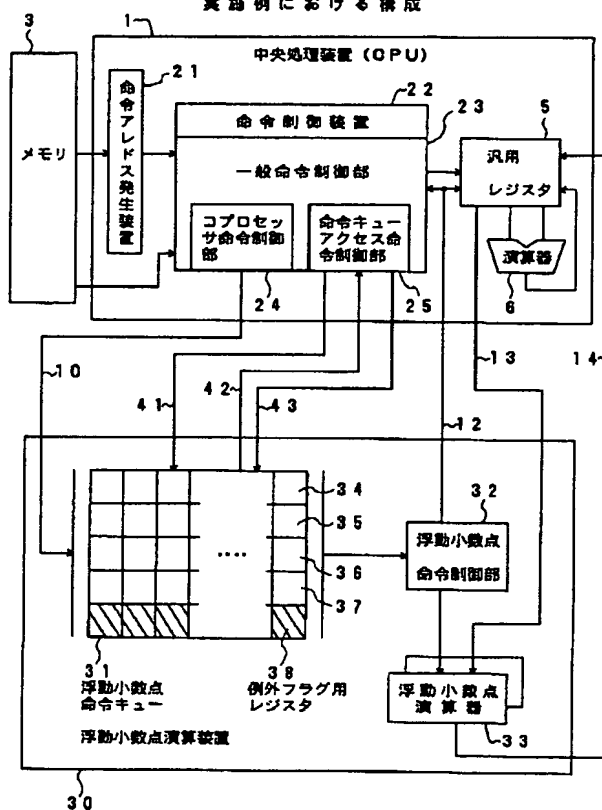
【図5】

コプロセッサの動作テスト処理方法



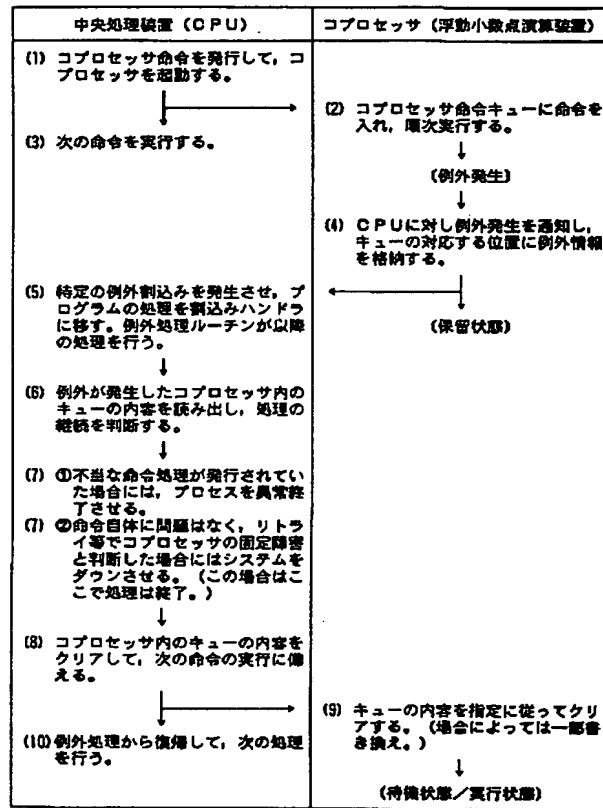
【図2】

実施例における構成



【図 4】

## 例外処理のフローチャート



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**